

第14章 面向对象的PHP编程

- 面向对象编程（OOP, Object-Oriented Programming）是一种编程思想，在很多现代计算机语言中都有面向对象编程的概念，不同的计算机语言，这一思想所表现的内容丰富程度是不完全相同的。例如，Java语言从一开始就设计成面向对象的编程语言，可以说面向对象就是其本质内涵。虽然PHP目前的版本中也支持面向对象的编程，但PHP并不是一个真正意义上的面向对象的编程语言，面向对象编程并不是PHP的优势所在。当然，在使用PHP的某些大型应用项目中加入面向对象的编程，而且在项目中只用类和对象，随着项目越来越大，面向对象编程可能会对项目有所帮助。因为，面向对象的编程代码比较容易维护和重用，有时，这甚至是一个Web项目的成败关键所在。
- 本章主要讲述PHP面向对象编程的抽象数据类型和数据封装等基本内容，这并不表明PHP中面向对象编程只有这些内容，有兴趣的读者可以深入了解多态等面向对象编程的高级部分。

14.1 面向对象编程的基本概念

- 要理解面向对象的编程，首先要理解什么是对象。在现实世界中，人们所面对的一切事物都可称之为对象。一本书、一个人、一架飞机、一只猫等都是实实在在的对象。每本书都会有一个名字、价格、作者等属性，这本书可以被出售，也可以被借阅。在面向对象的编程的过程中，首先要确定的就是对象，以及各对象之间的相互关系。
- 假设，在程序中确定了书这一对象，它包括的数据有：书名、价格、作者、出版社及出版数量等。在书这个对象上可能进行的操作有：查看书的名称（价格、作者、出版社等）、出售这本书或借阅这本书。这个简单的举例说明了在面向对象的编程中，对象是由数据以及在数据上进行的操作组成的。从抽象的角度来看，方法是一个对象的所做的动作，而属性是对象的特性。从编程角度来看，方法就是函数，而属性就是变量。面向对象编程的最基本概念包含以下3个特性。
 - 抽象数据类型和数据封装。
 - 继承。
 - 多态。



14.2 PHP中的类和对象

- PHP支持面向对象的编程，支持类和对象的概念。从数据类型的角度看，对象是一种比较特殊的数据类型。它由一个事先定义好的类生成，而类由用户自己定义，它由一系列数据和对这些数据操作的数个函数组成。
- 可以这么认为，类是一种用户自定义的数据类型，通过这个类型可以定义一个该类型的变量，这个变量就是该类型的对象。通过下面的代码，可以使读者对类和对象有一个感性上的认识。
- `$a_man = new person;`
- 上面的代码，类是person，而\$a_man就是person类型的变量，即\$a_man是类person的对象。



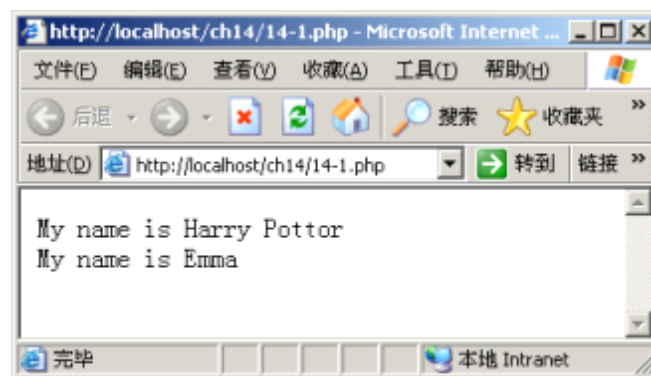
14.2.1 类的创建

- 在PHP中使用下面的语法定义类：
- `class classname`
- `{`
- `statement`
- `}`
- `class`是PHP的保留关键字，表示开始类的定义。`classname`是类名，由开发人员指定。由花括号“`{`”和“`}`”扩住的语句`statement`是类的实体，它一般由数据和对这些数据进行操作的函数组成。类中的数据，一般被称作成员变量（也叫属性），类中的函数一般被称作成员函数（或称作方法）。下面的代码定义了一个类。
- `class person`
- `{`
- `private $name;`
- `}`



14.2.2 类的实例化——对象

- 类只是提供了一种数据类型的模板，它本身并不能做具体的某种数据处理。只有将类具体化、实例化，才可以完成数据处理操作。类的实例化就是前面提到的对象，对象是程序执行过程中的类的实体。
- 在PHP中建立好一个类后，就可以使用 new运算符生成一个类的实例，即创建该类的一个对象。

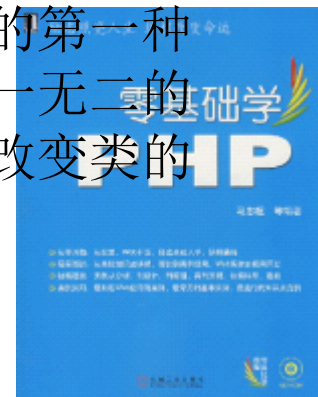


在程序中使用对象



14.2.3 构造函数和析构函数

- 在一个类中声明一个名为 `__construct` 的函数，这个函数称作构造函数，它在建立一个对象实例时被自动调用。注意，关键字 `construct` 之前的 `__` 是两个下划线。就像其他任何函数一样，构造函数可以带有参数。通常，在构造函数中使用这些参数完成对象属性的初始化等操作。在类中定义一个名为 `__destruct` 的函数，它称为析构函数，PHP将在对象被销毁前自动调用这个函数。
- 另外一种命名构造函数和析构函数的方式是，使用类的名称。比如一个名叫 `Cat` 的类，在类定义时命名一个名叫 `Cat()` 的函数，那么这个函数就是构造函数。刚才介绍的第一种声明构造函数的方法，可以使构造函数有一个独一无二的名称，而不论它所在的类的名称是什么。这样在改变类的名称时，就不再需要改变构造函数的名称。



14.2.4 继承

- 对于面向对象的编程，通过继承可以增加或重写类的方法。这意味着，继承类可以指定更多的属性或方法，而且允许继承类访问基类的方法。继承体现了“is_a”的关系，即“是一个”的关系。例如，Person是一个类，类Worker类继承自类Person，那么可以说，Worker类也是Person类。在PHP中，可以通过关键字extends从一个类派生出继承类。



14.2.5 访问对象的属性和方法

- 一个对象实例的属性是变量，就像 PHP 的其他变量一样，只不过必须使用 `->` 运算符来引用它们，但不需要在属性前使用符号 `$`。访问方法和访问属性一样，使用 `->` 运算符用来指向实例的方法。对象的方法执行起来和普通函数几乎相同。
- 在前面讲述类的继承时讲到，如果一个类从另一类中继承而来，基类中的属性和方法将在派生类中都有效。即使在派生类中没有声明，基类中的方法和属性一样会在派生类中有效。如果要访问一个继承的属性，只需像访问基类自己的属性那样引用即可，使用 `::` 运算符。在 PHP 的类继承用法中，有两个特殊的命名空间：`parent` 和 `self`。`parent` 命名空间指向父类，`self` 命名空间指向当前类。



14.2.6 PHP中类的静态成员

- 静态成员包括静态方法和静态属性。类的静态成员与类的一般成员不同，静态成员与类的实例无关，只与类本身有关。这意味着，类的静态属性可以由该类所有的对象所共享，它非常类似于函数的全局变量，只是它只属于某个固定的类，并且有访问限制。类似地，静态方法也与特定的对象无关，它非常类似于全局函数。静态方法可以完全访问类的属性，也可以由对象的实例来访问。
- 在实际开发中，如果希望在不存在有效对象的时候调用一个方法，那么就可以使用静态方法。



14.2.7 PHP中一些和类有关的函数

- 本节介绍一些和类有关的 PHP函数，这些函数在 PHP中被称为Magic Methods，因为它们在PHP是比较特别的，使用它们可以完成很多功能。这些函数如下所示。
- `__construct()`：构造函数，当实例化一个类对象时调用。
- `__destruct()`：析构函数，当一个对象不再使用时调用。
- `__get()`：当访问某个类没有显示定义的属性时，该函数被调用。
- `__set()`：当设置一个不存在的属性性时，该函数被调用。
- `__call()`：当访问一个不存在的方法时，该函数被调用。
- `__toString()`：将一个对象转换成字符串。
- `__clone()`：克隆一个对象时使用。



14.3 实例分析——深入理解PHP的类和对象

- 有了前面学习关于面向对象编程的基础知识，本节通过一个具体实例来深入认识PHP中的类和对象。在面向对象编程的思想中，任何一个事物都是对象，一个Web页面也不会例外。这节就使用PHP创建一个代表Web页面的类，使用该类可以生成HTML页面。首先分析一个Web页面应该具备哪些基本属性。



14.4 小结

- 本章讲述了PHP中的面向对象编程的有关概念，以及通过简单明了的示例代码，演示了如何在PHP中进行面向对象编程。其中，涉及到的面向对象的概念如下所示。
- 什么面向对象编程。
- PHP中类和对象的概念。
- 如何在PHP程序中使用类和对象。
- 构造函数和析构函数。
- PHP中继承的概念和使用。
- PHP中一些和类有关的函数。
- 最后通过分析一个示例，进一步讲述面向对象编程在PHP中的应用。

